Virtual Laboratory for Modeling Systems: A Sustainable Approach

Antoni Grau, Yolanda Bolea
Automatic Control Dept
Technical University of Catalonia, UPC
Barcelona, Spain
{antoni.grau,yolanda.bolea}@upc.edu

Abstract –**This paper discusses, first, the use of a research project as a didactic material for laboratory practices. The objective of TOLLA project (http://tolla.upc.edu) is to develop ecological models for simulate the behaviour of environmental variables (phytoplankton, zooplankton, nutrients, egestion…) in a set of ponds in the Technical University Campus. This experience will be shifted to the classroom in form of laboratory practices for the subjects related with simulation and modelling in different engineering degrees, because we have the belief that undergraduate students must receive an Education for the Sustainability independently of their career. The paper also discusses the use of Easy Java Simulations software to implement the models, their views and the simulations, in an easy-to-use and interactive programming environment that generates standalone Java applications or applets, or in conjunction with Matlab/Simulink.**

I. INTRODUCTION

In our modern society, distance education has become a viable solution for students who require more flexible, accessible, and adaptive teaching systems, without spatial and temporal restrictions [1], [2]. In the past, the interaction methods for distance education were limited to the telephone, postal mail, or fax. Today's new information technologies provide alternative tools for improving teacher-student interaction, two of which can be pointed to as the most capable and reliable for distance education. These tools are hypermedia systems as a new way of arranging information and wide-area communication networks (i.e., the Internet) for information support [3].

Although these tools are sufficient for constructing support systems for subjects without a strong practical component, teaching of systems modeling or other subjects with strong experimental content requires a new element. This new element must allow students to apply the knowledge acquired in a way that goes beyond the traditional physical laboratory, which requires the presence of students as well as an instructor or tutor. If the laboratory environment is to be transferred to distance education, the element required to put simulation and modeling concepts into practice is the virtual laboratory [4][5][6].

Today, numerous commercial and university computer tools are available for the analytical study of dynamic systems from a purely graphic or a numerical viewpoint, or a combination of both approaches [7][8][9]. These quantitative tools require the user to have a university-level knowledge of mathematics and a solid background in control systems.

This article describes the experience at the Automatic Control Dept at the Technical University of Catalonia (UPC) in teaching "modeling and control of systems" subjects that uses dynamic and interactive simulations in a stand-alone or Web-based environment. The article focuses on how this new stand-alone experimentation environment maintains a clear separation between the graphical experimentation interface, developed in Java, and the math and simulation engine. By constructing the environment in this fashion, the math engine can be replaced with a different one or with a real plant, or can even be ported to a remote server. A Web-based, multi-user virtual lab is also possible without the necessity of reprogramming the experimentation interface code. Other differences with respect to other tools are the dynamic simulations, the user interactivity, the generation of new experiments as goals change, and the opportunity to practice different models for the same problem. The article also focuses in a new way of understanding the contents of technical subjects, such as modeling and control of dynamic systems, following an strategy well-implanted at the UPC: incorporating the education for the sustainable development (ESD) in the curriculum.

The article is organized as follows. First, the inherent relationship between the environment and ecosystems and the modeling and control of systems in the actual world is explained. Second, the TOLLA project is briefly introduced. This research project leads to implement virtual laboratory practices of ecosystems modeling based on a real case study, and one of these models is described. *Easy Java* Simulations is the tool that will be used to program the model and implement the views for the simulation. Finally, the didactic methodology used for the laboratory is explained, analyzing the task for the teaching as well as the students' tasks.

II. MODELING AND CONTROL OF SYSTEMS FOR THE SUSTAINABILITY

In the last years society has become aware of that the pattern of current development affects our Planet in a noxious way. The wrong and excessive use of the natural resources, the group of substantial changes in the environment (deforestation, natural disasters, climatic, pollution...) has generated the environmental problem that all the countries of the world are living, especially the industrialized ones. For this reason the XXIst century outlines a new and important challenge in the society: the sustainability.

The University plays an important role in transmitting this environmental conscience in its studies traditionally related with the natural environment such as biology, medicine, ecology, zoology, but its interest goes beyond this point and arrives to the merely technological subjects [10]: engineering

1-4244-0136-4/06/$20.00 ©2006 IEEE

(industrial, civil, chemical, automatic control, telecommunications, etc.) and computer science. Following this path, the Technical University of Catalonia, UPC [11] (among other universities [12]) gives priority to the environmental education in its curricula, mainly technological. The reason lies in the belief that many problems in our Planet can be solved, achieving the so-called "sustainability", through actions indirectly related with the environment using the technological advances not only in a economic and efficient productive manner but also taking into account our natural environment [13][14].

The application's domains of biological and ecological systems provide many opportunities and challenges for the technologies of modeling and control [15][16]. High schools are places where students and teachers share knowledge, learn skills, and transmit values. Specifically, Control Engineering degree opens up possibilities to connect the network analysis methodology to traditional control theory, to reintroduce a network-based control parameter using flow analysis and to extend the methodology to network storage analysis [17]. Model ecosystems are constructed and used to investigate these properties. For this purpose we propose to introduce terms of sustainability in the subject *Simulation and Modeling of Dynamic Systems* of this degree, mainly by the presentation of environmental case-studies systems [18]. Normally, the main objectives of environmental modeling are the following: 1) prediction and estimation of important environmental variables using a composite model to explore efficient environmental strategies and management schemes; 2) prevention of environmental risk, this area involves work on accident prevention in industry, through accident modeling and evaluation of the possible effects; 3) the challenge of environmental control issues for industry (chemical industry, iron and steel industry, etc.). Over the past decade, industry in general has made efforts to save energy, improve efficiency of energy use, reduce CO_2 emission, and promote recycling and life cycle assessment.

III. TOLLA PROJECT

The fact that students use a virtual laboratory, it does not suppose that the laboratory practices have to be invented, and the idea is just the opposite: we propose the use of practices based on a real case. This can stimulate students in their practices and increment their interest in an environmental application in order to aware them that environment and engineering are closely related. The role of these students in the professional world will be decisive for the preservation of the natural resources.

The REAL project (research for environmental excellence) is an initiative started in 2001 inside the 2n Environmental Plan at the UPC, to provide the University of a Campus model based on the environmental excellence.

The construction of a new Campus in Castelldefels, near Barcelona, has generated a perfect setting to develop research activities in Environment and Sustainable Development. Furthermore, REAL promote the interdisciplinary nature of the participating research groups. Regarding this subject, our research group at the Automatic Control Dept has been leading the TOLLA project.

TOLLA project (http://tolla.upc.edu) is a research project which objective is the study, prediction, management and control of the water and life quality in a set of ponds located in the aforementioned Campus, fig 1. Such a management and control require data acquisition, storage and analysis, in order to forecast the behavior of different variables of interest and act on them in order to maintain a sustainable system. Our task consisted in finding an ecological model to simulate the evolution of the life (zooplankton and phytoplankton) and nutrients in the ponds which are subjected to severe ecological disturbances due to the proximity to a big industrialized city such as Barcelona.

Fig 1. Aerial view of the ponds in the Campus.

The proposed model is:

$$\frac{dPhy}{dt} = growth_{Phy} - egestion - mort_{Phy}$$
$$\frac{dZoo}{dt} = (1-\gamma)\cdot egestion - mort_{Zoo}$$
$$\frac{dN}{dt} = \gamma \cdot egestion + mort_{Zoo} + mort_{Phy} - growth_{Phy}$$

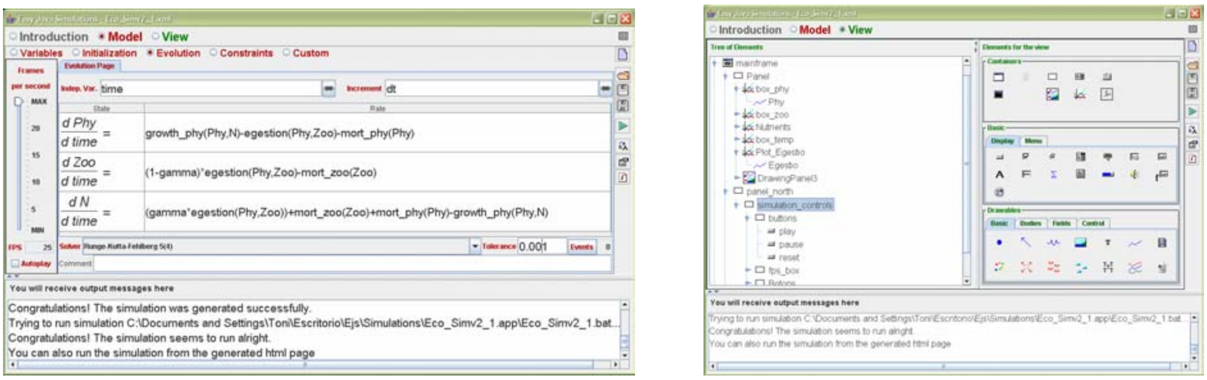
This model, that we call PZN [19], is a system of ordinary differential equations (ODEs), where *Phy* is the phytoplankton and *Zoo* is the zooplankton in the ecosystem expressed in [mmol m^{-3}], and *N* is the amount of available nutrients in [mmol m^{-3}], too.

$$growth_{Phy} = V_m \cdot f(T)\cdot f(N)\cdot Phy$$
$$egestion = R_m \cdot (1-e^{-\Lambda Phy})\cdot Zoo$$
$$mort_{Phy} = m \cdot Phy$$
$$mort_{Zoo} = g \cdot Zoo$$

where $growth_{Phy}$ is the phytoplankton growing variable, *egestion* indicates the opposite of ingestion (zooplankton eats phytoplankton), $mort_{Phy}$ and $mort_{Zoo}$ indicate the mortality of phytoplankton and zooplankton, respectively, and expressed in [mmol m^{-3}·day^{-1}].

TABLE I
PARAMETERS USED IN THE ECOLOGICAL MODEL

Name of parameter	Description	Value	Unities
g	Zooplankton mortality coefficient	0.2	[day^{-1}]
T	Water temperature	Variable	ºC
k_N	Half saturation constant for nitrogen	0.2	[mmol m^{-3}]
m	Phytoplankton mortality coefficient	0.1	[day^{-1}]
R_m	Herbivore maximum ingestion index	1.0	[day^{-1}]
V_m	Phytoplankton maximum growing index	2.0	[day^{-1}]
γ	Zooplankton ingestion efficiency	0.3	--
Λ	Herbivore grazing constant	0.75	[m^3 mmol^{-1}]



Indep. Var. time   Increment dt

$\frac{d\ Phy}{d\ time}$ = growth_phy(Phy,N)-egestion(Phy,Zoo)-mort_phy(Phy)

$\frac{d\ Zoo}{d\ time}$ = (1-gamma)*egestion(Phy,Zoo)-mort_zoo(Zoo)

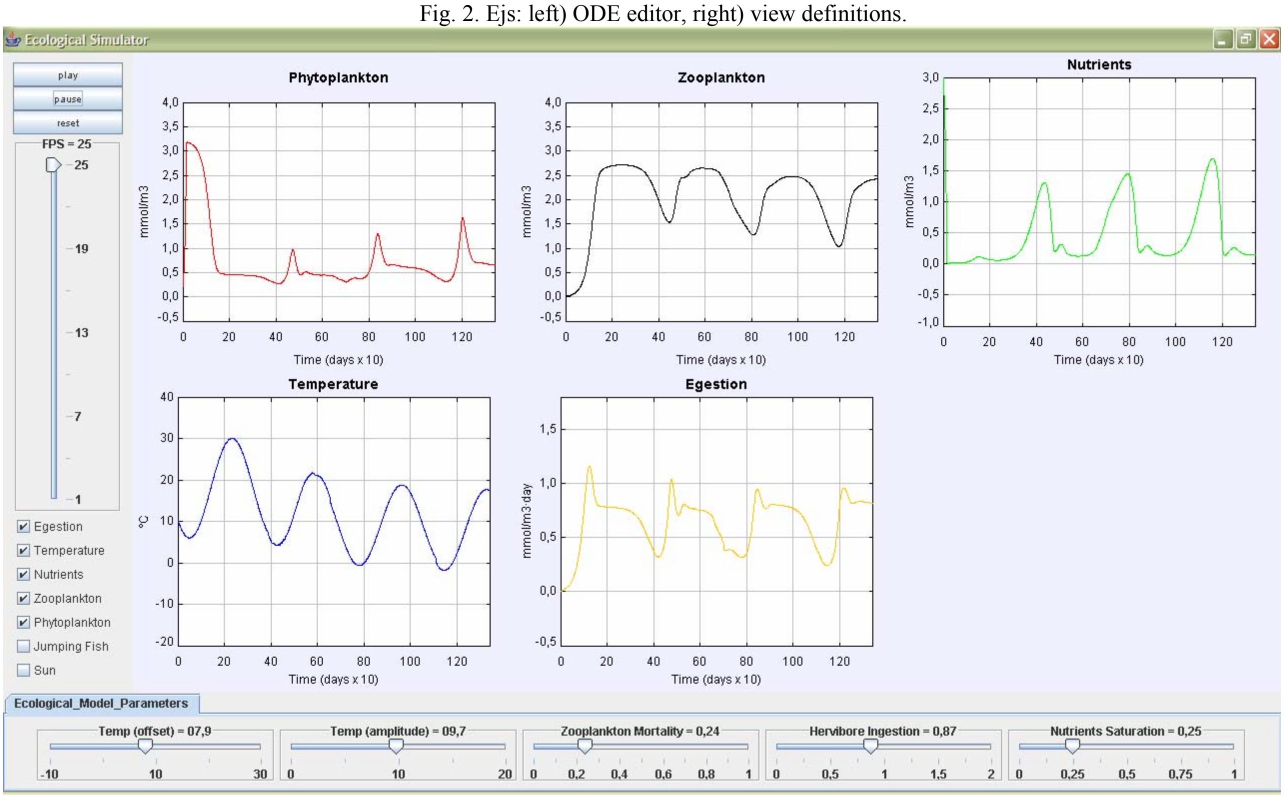
$\frac{d\ N}{d\ time}$ = (gamma*egestion(Phy,Zoo))+mort_zoo(Zoo)+mort_phy(Phy)-growth_phy(Phy,N)

Solver Runge-Kutta-Fehlberg 5(4)   Tolerance 0.001

```
You will receive output messages here
Congratulations! The simulation was generated successfully.
Trying to run simulation C:\Documents and Settings\Toni\Escritorio\Ejs\Simulations\Eco_Simv2_1.app\Eco_Simv2_1.bat...
Congratulations! The simulation seems to run alright.
You can also run the simulation from the generated html page
```

```
You will receive output messages here
Trying to run simulation C:\Documents and Settings\Toni\Escritorio\Ejs\Simulations\Eco_Simv2_1.app\Eco_Simv2_1.bat
Congratulations! The simulation seems to run alright.
You can also run the simulation from the generated html page
```

Fig. 2. Ejs: left) ODE editor, right) view definitions.



Temp (offset) = 07,9
Temp (amplitude) = 09,7
Zooplankton Mortality = 0,24
Hervibore Ingestion = 0,87
Nutrients Saturation = 0,25

Fig. 3. Simulation according to the ODE of the model and the defined view.

Phytoplankton growth (*growth$_{Phy}$*) is function of temperature $f(T)$ and dissolved nitrogen as nutrients $f(N)$.

Temperature function is: $f(T) = \log 2\,(0.851)(1.066^{T})$ derived from [20].

Using the Michaelis-Menten formula [19], nutrients function is:

$$f(N) = \frac{N}{k_N + N}$$

where k_N is the half saturation constant for nitrogen uptake.

The different parameters used in the model are described in table I.

IV. EASY JAVA SIMULATIONS

Easy Java Simulations is a freeware, open-source tool developed in Java, specifically designed to create interactive dynamic simulations [21].

Ejs was originally designed to be used by students for interactive learning under the supervision of educators with a low programming level. However, the user needs to know in detail the analytical model of the process and the design of the graphical view. *Ejs* guides the user in the process of creating interactive simulations, in a simple and practical way.

The architecture of *Ejs* derives from the model-view control paradigm, whose philosophy is that interactive simulations must be composed of three parts: the model, the view, and the control. According to that, the steps to build an application in *Ejs* are the following: (1) To define the model is necessary to specify the variables that describe the system and the mathematical equations interrelating them; (2) define the view in order to represent the states of the process; and (3) define the control in order to describe the actions that the modeler can execute above the simulation.

These three parts are interconnected because the model affects to the view and the control actions affect to behaviour of the model. Finally, the view affects to the model and to the control because the graphical interface can contain information about them.

Ejs implements its particular procedure to create specific interactive simulations since it removes the control part. Consequently, *Ejs* simulations are created through specifying a model to be run by the *Ejs*'s simulation engine and by building a view to visualize the model state and that readily responds to user interactions.

So, to define the model in *Ejs*, it is necessary to identify the variables that describe the process, to initialize them, and also to describe the mathematical equations that generate the model.

The view is now the user-to-model interface of *Ejs* interactive simulations. It is intended to provide a visual representation of the model's relevant properties and dynamic behavior and also to facilitate the user's interactive actions. *Ejs* includes a set of ready-to-use visual elements. With them, the modeler can compose a sophisticated view in a simple, drag and drop way. The properties of the view elements can be linked to the model variables, producing a bi-directional flow of information between the view and the model. Any change of a model variable is automatically displayed by the view. Reciprocally, any user interaction with the view automatically modifies the value of the corresponding model variable.

To summarize, the model is the scientific part of the simulation; yet the creation of the necessary graphical user interface (the view) is the part of the simulation that demands more knowledge of advanced programming techniques.

Once the modeler has defined the model and the view of the interactive simulation, *Ejs* generates the Java source code of the simulation program, compiles the program, packs the resulting object files into a compressed file, and generates HTML pages containing the narrative and the simulation as an applet. The user can readily run the simulation and/or publish it on the Internet. Easy Java Simulations, the software tool, a complete English manual for it, can be downloaded for free from *Ej*s' web server at [22].

Developers can then insert Matlab code in an *Ejs* application to make calculations in the Matlab workspace. But also, the model of a system can be completely or partially developed using just Simulink block diagrams. In this last situation, *Ejs* can just be the tool that will help create the view of the simulation, i.e., the visualization and user interface for the Simulink model. This process has been made as simple as possible. Once a Simulink simulation has been selected in *Ejs* as part of the model, all the Simulink variables can be accessed from *Ejs* and be associated to regular variables. So, developers can link them to the graphical elements like regular Ejs variables. When running the simulation, *Ejs* will automatically take care of running the Simulink model and of the two-way exchange of information between the model and the view (on the side of Ejs) and the Matlab workspace.

V. VIRTUAL LABORATORY

The laboratory practices that we propose in the *Simulation and Modeling of Dynamic Systems* subject consist in the use of a virtual laboratory based on the Easy Java Simulations tool. The objective of this laboratory is to implement and test models for the described ponds.

Teaching staff have an initial task of preparing the contents of the practice, and it consists in the following steps:

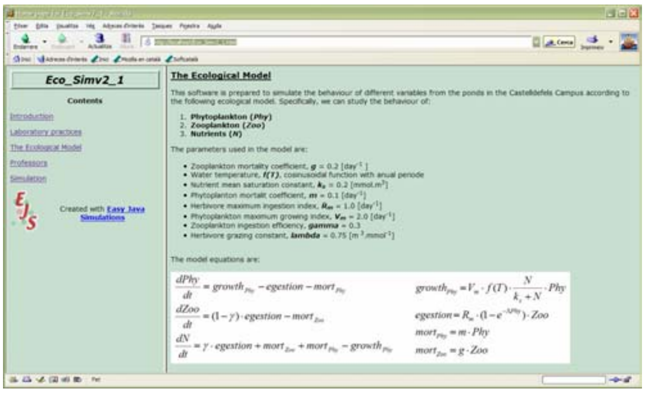
1. To develop the model of the real system by physical or empirical equations.
2. To implement the model with the *Ejs* in order to verify the equations, and propose a specific view.
3. When the above steps are perfectly done and the view is attractive enough, it is necessary to create a web page with the output of *Ejs*. This web page will contain descriptive information about the practice, the objective, the methodology, the deadline and other academic information as well as the simulation results. These results have a great interest for the

student because he/she can check what are the expected results for the proposed practice.

Once the practice is on the web, students can begin to do the practice in a local way. This aspect is very important because *Ejs* can be easily installed in their own PC and it is not necessary to be connected to Internet to do the job. Ejs can be installed for the most common operating systems.

Then, the steps the student must follow are:

1. Read attentively the practice. Previously, *Ejs* is introduced in the classroom and some exercises have been realized. In fig. 4 it can shown the practice formulation. Due to the *Ejs* interactivity, students are request to show the more significant variables in the model (in this case phytoplankton, zooplankton, nutrients, and egestion) as well as to allow the modification of the most significant parameters of the model (temperature, zooplankton mortality, phytoplankton mortality, herbivore ingestion, nutrients saturation, among others). With a set of sliders these values can be easily modified, causing an sudden change in the simulation.
2. Implement the model with *Ejs*, following the ODE system proposed for the practices' instructor. Implement a view for the simulation results. Students have, as example, the results that instructor has posted in Internet, fig. 5. It is not necessary *Ejs* installation to see the simulation execution because it is just a Java applet.
3. When the student finishes the implementation and the results are similar to the solution, he/she has to send the file where *Ejs* stores all the model equations, views and description. This information is stored in a XML file, and its size is relatively small. This is another feature of *Ejs*, the information is not in binary format nor codified in any way.
4. The instructor can easily evaluate the task of the student just compiling the XML file and checking the simulation results.



The Ecological Model

This software is prepared to simulate the behaviour of different variables from the ponds in the Castelldefels Campus according to the following ecological model. Specifically, we can study the behaviour of:

1. **Phytoplankton (*Phy*)**
2. **Zooplankton (*Zoo*)**
3. **Nutrients (*N*)**

The parameters used in the model are:

- Zooplankton mortality coefficient, **g** = 0.2 [day^{-1}]
- Water temperature, ***f(T)***, cosinusoidal function with anual periode
- Nutrient mean saturation constant, $\boldsymbol{k_s}$ = 0.2 [$mmol.m^3$]
- Phytoplanton mortalit coefficient, **m** = 0.1 [day^{-1}]
- Herbivore maximum ingestion index, $\boldsymbol{R_m}$ = 1.0 [day^{-1}]
- Phytoplankton maximum growing index, $\boldsymbol{V_m}$ = 2.0 [day^{-1}]
- Zooplankton ingestion efficiency, **gamma** = 0.3
- Herbivore grazing constant, **lambda** = 0.75 [$m^3.mmol^{-1}$]

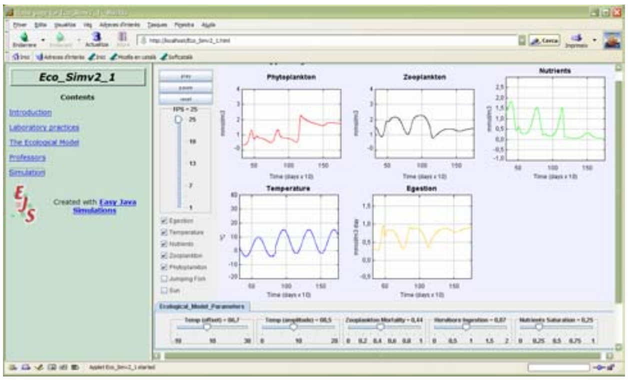
The model equations are:

$$\frac{dPhy}{dt} = growth_{Phy} - egestion - mort_{Phy} \qquad growth_{Phy} = V_m \cdot f(T) \cdot \frac{N}{k_s + N} \cdot Phy$$

$$\frac{dZoo}{dt} = (1-\gamma) \cdot egestion - mort_{Zoo} \qquad egestion = R_m \cdot (1 - e^{-\lambda Phy}) \cdot Zoo$$

$$\frac{dN}{dt} = \gamma \cdot egestion + mort_{Zoo} + mort_{Phy} - growth_{Phy} \qquad mort_{Phy} = m \cdot Phy \qquad mort_{Zoo} = g \cdot Zoo$$

Fig. 4. Web page: model formulation for students.



☑ Egestion
☑ Temperature
☑ Nutrients
☑ Zooplankton
☑ Phytoplankton
☐ Jumping Fish
☐ Sun

Fig. 5. Web page: simulation results.

Using this method, two of the current problems in distance teaching are solved: temporary availability and the training aspect. Students/operators can practice anywhere at any time, without the need to go to a training centre or keep to a timetable (the teaching/training system will be available 24 hours a day via their computers).

From the teaching staff point of view, this method allows the addition of new practices in Internet as well as new simulations results that students have to achieve.

Apart from these practices, it is proposed to students (as extra work) to change the simulation engine from Ejs to Simulink, using the views with Ejs. This shift can be easily (but also carefully) done and it has some academic interest, because it demonstrates how to link different applications. Ejs accepts also to use the MATLAB workspace to share variables, using it as math engine.

VI. CONCLUSIONS

The introduction of sustainable skills in this regular subject has a good acceptation among the students as well as the department teaching staff. Everybody realizes the importance of ESD in the University's subjects. The assessment of these skills is not easy because they have been transmitted as experiences and as attitudes. The real effect we hope that it will be achieved in a long term period.

The modeling of control systems teacher is an expert in his/her subject but not in programming techniques. The development of interactive didactic applications is not so easy with the typical tools that a modeler teacher uses to explain concepts, as for example, Matlab/Simulink. At the same time, there are many commercial tools designed to create animations and multimedia but, in general, scientific simulations cannot be created with them because they make a poor or inexistent use of the mathematical language to define the system to model.

Ejs should be considered as a first step in the development of further and more advanced tools intended for control engineering educators with some knowledge of programming, not for advanced programmers with little knowledge of control and modeling. Ejs has been conceived for people who are more interested in the contents of the simulation, the simulated phenomenon itself (study and analysis of the system behavior), and much less in the technical aspects needed to build the simulation.

Hence, *Ejs* provides a conceptual structure and simplified tool that allows concentrate most of the time in the description of the model of the system to simulate. Ejs facilitates the creation of the user interface, and Matlab/Simulink let teachers develop the model using the language they are familiar with.

IV. ACKNOWLEDGMENT

Authors want to thank to Marta Pujadas and all the staff at CITIES, UPC, for their kind help in collecting REAL project data, pictures and other related information.

V. REFERENCES

[1] B. Aktan, C.A. Bohus, L.A. Crowl, and M.H. Shor, “Distance learning applied to control engineering laboratories,” *IEEE Trans. Educ.,* vol. 39, no. 3, pp. 320-326, 1996.

[2] S.E. Poindexter and B.S. Heck, “Using the Web in your courses: What can you do? What should you do?” *IEEE Contr. Syst. Mag.,* vol. 19, pp. 83-92, Feb. 1999.

[3] K. Maly,H. Abdel-Wahab, C.M. Overstreet, J.C.Wild, A.K. Gupta, A.Youssef, E. Stoica, and E.S. Al-Shaer, “Interactive distance learning over intranets,” *IEEE Internet Comput.*, vol. 1, no. 1, pp. 60-71, 1997.

[4] J.B. Patton and P. Jayanetti, “The making of multimedia power systems control and simulation labware,” *IEEE Trans. Educ.*, vol. 39, no. 3, pp. 314-319, 1996. Available: http://www.ece.gatech.edu/users/192/web-use

[5] C. Schmid, “A remote laboratory using virtual reality on theWeb,” *Simulation,* vol. 73, no. 1, pp. 13-21, 1999.

[6] J. Sánchez, F. Morilla, S. Dormido, J. Aranda, and P. Ruipérez, “Conceptual learning of control by Java-based simulations,” in *Proc. IEEE/IFAC Symp. Advances in Control Education* (ACE’2000), Gold Coast, Australia, 2000.

[7] B.Wittenmark, H. Haglund, and M. Johansson, “Dynamic pictures and interactive learning,” *IEEE Contr. Syst. Mag.,* vol. 18, pp. 26-32, June 1998.

[8] R.C. Garcia and B.S. Heck, “Enhancing classical control education via interactive GUI design,” *IEEE Contr. Syst. Mag.,* vol. 19, pp. 77-82, June 1999.

[9] TOPAS by ACT GmbH [Online]. Available: http://www.act-control.com/topas.pdf

[10] P. Weaber, et al. *Sustainable Technology Development,* GreenLeaf, 2000.

[11] http://www.upc.edu/mediambient

[12] http://www.odo.tudelft.nl/english/index.html

[13] I. Capdevila, *L’ambientalització de la universitat*, Col·lecció Monografies d’Educació Ambiental*,* núm.6, Ed. SCEA, Departament de Medi Ambient-UPC, 1999.

[14] L. Chauveau, *Ecological Risks, an avoidable threat?,* Larousse, 2004.

[15] B. D. Fath, et al. “Distributed control in ecological networks”, *Ecological Modelling*, vol.179, pp.235-245, 2004.

[16] Y. Hashimoto, et al., “Control approaches to bio-and ecological systems”, *Control Engineering Practice,* vol.12, pp.595-603, 2004.

[17] A. Grau, Y. Bolea, J. Domingo, J. Gámiz and H. Martínez, *Sustainability and Automatic Control*, Sustainability and Technology Collection, no. 1, Edicions UPC (in catalan), 2005.

[18] A. Okubo and S.A. Levin, *Diffusion and Ecological Problems,* Springer, 2001.

[19] A. Chapelle et al., “Modelling nitrogen, primary production and oxygen in a Mediterranean lagoon. Impact of oysters farming and inputs from the watershed”, Ecological Modelling, vol.127 , pp. 161–181, 2000.

[20] R.W. Eppley, “Temperature and phytoplankton growth in the sea”, *Fish. Bull*, vol.70, pp.1063-1085, 1972.

[21] F. Esquembre, “Easy Java Simulations: A software tool to create scientific simulations in Java”, Comp. Phys. Comm. 156 (2), pp 199-204, 2004.

[22] Easy Java Simulations’ home page at http://fem.um.es/Ejs.